AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1. (Currently Amended) A computer implemented method of identifying equivalent portions of one or more unsorted hierarchically-organized data structures, the method comprising:

collapsing plural nodes of the one or more unsorted hierarchically-organized data structures ~~thereof~~ into respective representations, [[that]] wherein each of the respective representations incorporate information of a respective node and information [[that]] of any child nodes of the respective node ~~thereof~~; and

based on correspondence of particular instances of the collapsed representations, identifying the respective portions as equivalent,

wherein the collapsing is order-insensitive with respect to information of the respective child nodes,

wherein the collapsed representations include respective aggregations of orthogonally-encoded child node information.

2. (Cancelled)

3. (Currently Amended) A method as recited in claim [[2]] 1,

wherein a unit of orthogonally-encoded child node information includes a power-of-two encoded mapping of a concatenation of the child node information with a similarly encoded mapping of respective information of child nodes thereof.

4. (Currently Amended) A method as recited in claim [[2]] 1,

wherein a unit of orthogonally-encoded child node information includes a power-of-two encoded mapping of a concatenation of the child node information with recursively encoded mappings of respective sub-hierarchies thereof.

5. (Original) A method as recited in claim 1,

wherein the order-insensitive collapsing includes an arithmetic sum of orthogonal binary encodings of child node information.

6. (Original) A method as recited in claim 1,
wherein distinct tables are defined for each level of the hierarchically-organized data structure.

7. (Original) A method as recited in claim 1,
wherein a table spans multiple levels of the hierarchically-organized data structure.

8. (Original) A method as recited in claim 1, wherein, at a particular node of the hierarchically-organized data structure, the order-insensitive collapsing includes:
an arithmetic addition of orthogonal binary encodings that identify corresponding table entries for respective child nodes; and
concatenation of a result of the arithmetic addition with an encoding of information for the particular node.

9. (Original) A method as recited in claim 8,
wherein the order-insensitive collapsing at the particular node further includes creating a new mapping of the concatenation, the new mapping being an encoding that is at least orthogonal with that for any other node at a same level of the hierarchically-organized data structure.

10. (Original) A method as recited in claim 8,
wherein the order-insensitive collapsing at the particular node further includes creating a new mapping of the concatenation, the new mapping being an encoding that is orthogonal with that for any other node of the hierarchically-organized data structure.

11. (Original) A method as recited in claim 8,
wherein the encoding of particular node information is a string encoding thereof.

12. (Previously Presented) A method as recited in claim 1,
wherein the correspondence of particular instances of the collapsed representations is based on identity of respective mapped codes.

13. (Original) A method as recited in claim 1,
wherein the order-insensitive collapsing includes an arithmetic addition of orthogonally-encoded values that index into a store of child node information.

14. (Original) A method as recited in claim 1,
wherein the hierarchically-organized data structure includes at least three levels of nodes; and
further comprising performing the collapsing at successive ones of the levels of the hierarchically-organized data structure.

15. (Original) A method as recited in claim 1,
wherein the hierarchically-organized data structure includes a tree-organized data structure.

16. (Original) A method as recited in claim 8,
wherein the hierarchically-organized data structure includes at least two levels.

17. (Original) A method as recited in claim 1,
wherein the hierarchically-organized data structure encodes subassembly information as sub-hierarchies thereof and encodes component parts at least at leaf nodes thereof.

18. (Currently Amended) A computer implemented method of identifying equivalent logical sub-trees of a tree-oriented data representation, the method comprising:
associating a first-level identifier with each of plural leaf nodes at a first-level of the tree, wherein distinct leaf node values are associated with distinct first identifiers and equivalent leaf node values are associated with same first identifiers; and

at each next level of the tree, associating an identifier with each node of a respective level ~~thereof~~, each such identifier including a current node contribution and a contribution associated with any child nodes of the current node ~~thereof~~,

wherein the child nodes contribution is computed using a combining function operative on identifiers associated with the child nodes,

wherein the identifiers and combining function are selected to satisfy orthogonality,

wherein the selected identifiers and combining function ensure that same combinations of child node identifiers result in same child nodes contributions irrespective of ordering of the child node identifiers, and

wherein for a second level of the tree, respective child nodes are the leaf nodes of the first-level of the tree.

19. (Original) A method as recited in claim 18,

wherein the identifiers are orthogonally-encoded mappings of respective string encodings of the current node contribution concatenated with respective orthogonally-encoded mappings of child node information.

20. (Currently Amended) A method as recited in claim [[18]] 19,

wherein the orthogonally-encoded mappings at each level of the tree-oriented data representation are in accordance with a corresponding level-specific table.

21. (Currently Amended) A method as recited in claim [[18]] 19,

wherein the orthogonally-encoded mappings for distinct portions of the tree-oriented data representation are in accordance with respective tables.

22. (Currently Amended) A method as recited in claim [[18]] 19,

wherein the orthogonally-encoded mappings for multiple levels of the tree-oriented data representation are in accordance with a single corresponding hash table.

23. (Currently Amended) A method as recited in claim [[18]] 19,

wherein the orthogonally-encoded mappings ~~hashes~~ for each level of the tree-oriented data representation are in accordance with a single corresponding table.

24. (Original) A method as recited in claim 18,
wherein, at least at any particular level of the tree-oriented data representation, the identifiers are orthogonally-encoded.

25. (Original) A method as recited in claim 18,
wherein the identifiers correspond to orthogonal binary encodings of integers; and
wherein the combining function includes addition.

26. (Original) The method of claim 18, employed in a duplicate elimination operation on the tree-oriented data representation.

27. (Original) The method of claim 18, employed in a duplicate identification operation on the tree-oriented data representation.

28. (Original) The method of claim 18, employed in an equality test operation on portions of the tree-oriented data representation.

29. (Currently Amended) A computer implemented method of representing hierarchically-organized data, the method comprising:
recursively collapsing sub-hierarchies of the hierarchically-organized data ~~thereof~~ using encodings, wherein the encodings ~~that, at least at a same level thereof,~~ include[[s]] orthogonal values;
representing any given node of the hierarchically-organized data as a concatenation of node-specific information with a combination of the orthogonal values for each collapsed sub-hierarchy beneath the given node ~~therebeneath~~.

30. (Original) The method of claim 29,
transforming from a first encoding of the hierarchically-organized data to a collapsed second form.

31. (Original) The method of claim 29,
employed to eliminate duplicate sub-hierarchies in the hierarchically-organized data.

32. (Original) The method of claim 29,

employed to collapse duplicate sub-hierarchies in the hierarchically-organized data,

wherein the concatenation further includes a count of duplicate sub-hierarchies collapsed beneath any given node.

33. (Currently Amended) A computer program product encoded in at least one computer readable medium, the computer program product comprising:

a program sequence including a recursively called set of instructions executable by one or more processors to operate on at least one instance of an hierarchically-organized data structure, the instructions, when executed, causing the processor to define a counterpart data structure by collapsing plural nodes of the hierarchically-organized data structure into respective representations, wherein [[that]] each of the respective representations incorporate information of a respective node and information [[that]] of any child nodes of the respective node ~~thereof~~, wherein the collapsing includes an order-insensitive aggregation of orthogonal encodings of information of the respective child nodes; and

an object implementing the counterpart data structure including at least one table wherein values of the at least one table ~~thereof~~ provide the orthogonal encodings and keys, wherein the keys ~~thereof~~ combine the information of respective nodes with an aggregation of the collapsed representations for child nodes of the respective nodes ~~thereof~~.

34. (Previously Presented) The computer program product of claim 33,

wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, or other communications medium.

35. (Currently Amended) An information management tool including software encoded in one or more computer readable media and executable by one or more processors, the information management tool comprising:

an encoding of a hierarchically-organized data structure instantiable in memory addressable by the one or more processors;

instructions executable by the one or more processors to operate on at least one instance of the hierarchically-organized data structure instantiated in memory, the instructions, when executed, causing the processor to define a counterpart data structure in the memory by collapsing plural nodes of the hierarchically-organized data structure into respective representations that each incorporate information of a respective node and information [[that]] of any child nodes of the respective node ~~thereof~~, wherein the collapsing includes an order-insensitive aggregation of orthogonal encodings of information of the respective child nodes.

36. (Original) An information management tool, as recited in claim 35, further comprising:

matching instructions executable by the one or more processors to identify distinct sub-hierarchies of the hierarchically-organized data structure as at least equivalent based on correspondence of the collapsed representations.

37. (Original) An information management tool, as recited in claim 35, further comprising:

matching instructions executable by the one or more processors to identify at least equivalent portions of first and second ones of the hierarchically-organized data structure based on correspondence of collapsed representations thereof.

38. (Original) An information management tool, as recited in claim 35,

wherein the order insensitive aggregation is performed recursively at successive levels of a collapsed sub-hierarchy.

39. (Original) An information management tool, as recited in claim 35, wherein the counterpart data structure includes:

at least one hash table; and

a recursively encoded mapping wherein, for any particular node of the hierarchically-organized data structure, a corresponding table entry encodes both respective values for child nodes thereof in accordance with the order-insensitive information and aggregation associated with the particular node itself, and

wherein, at least for same-level nodes of the hierarchically-organized data structure, corresponding values are orthogonal.

40. (Original) An information management tool, as recited in claim 35,
wherein the hierarchically-organized data structure encodes and a sub-assembly decomposition of a product configuration; and
wherein the information management tool further identifies, based on correspondence of collapsed representations of the hierarchically-organized data structure, equivalent sub-assemblies without regard to ordering of elements thereof.

41. (Original) An apparatus comprising:
a processor and memory addressable thereby; and
means for performing an element order independent comparison of hierarchically organized data structures using a transformation operation that orthogonally and recursively encodes child node information.